DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF TORONTO

CSC318S

THE DESIGN OF INTERACTIVE COMPUTATIONAL MEDIA

Lecture 10 — 11 Feb. 1998

INTERACTIVE DIALOGUE STYLES AND TECHNIQUES 1

Ronald Baecker
Professor of Computer Science,
Electrical and Computer Engineering, and Management
University of Toronto

10.1 A model of interactive dialogues

Content, subject matter of the dialogue
- The domain of discourse
- The person's task (Need for task analysis)
- Linear, "real time," as in command and control, versus non-linear, exploratory, as in problem solving, CAD

Context
- Constraints on the system and dialogue (hardware to be used, development time and cost, marketing requirements such as cost of system, etc.)
- Requirements on the task (speed, accuracy, urgency, etc.)

One partner in the dialogue — the person
- Intelligence
- Training
- Expertise, a product of intelligence and training (novice or expert)
- Frequency of use (regular or casual)
- Motivation or alienation
- Style (active or passive)
- Involvement (ultimate user or intermediary)

Other partner in the dialogue — the machine
- Response latency
- Computational bandwidth
- Response time

Output media, technologies, and devices
- Visual: B&W, colour, resolution, update bandwidth, etc.
- Auditory: Speech, non-speech audio, etc.

Input media, technologies, devices and actions
- Touch, speech, eye movement, etc.
- Typing, pointing, drawing, etc.

10.2 Design criteria for interactive dialogues

Consistency
Clarity
System must be *articulate*
System must facilitate *articulate expression*
Concept of trade-offs

10.3 Interaction paradigms and styles

Command names and simple command languages
Query languages and conversational programming languages
Natural language input
Voice input
Menus
Form filling, e.g., style sheets (a la Xerox Star)
Icons
Windows (tiled and overlapping)
Direct manipulation, WYSIWYG (What You See Is What You Get)
Graphical and gestural interaction, tablet and mouse dialogues
Multi-media interaction
3D interaction
Programming: Textual programming, visual programming, programming by example, programming by constraints

10.4 Common issues re interaction techniques

Who's in control? User or system? Or *mixed initiative*?
“Artificial languages,” and their lexical, syntactic, pragmatic, and semantic structure
Universal operators and *generic* commands
The role of *metaphor* (e.g., the desk top analogy)

10.5 Command names & simple command languages

User-initiated
- Harder for beginner, more efficient for expert
- Demands good retention by casual, infrequent users

User must remember command *sequence* for desired task

User must remember command *names* for desired subtask
- Difficulties in choosing “best, most natural” command name
- Designers have difficulty choosing “best” name
  - Likelihood that any two individuals would generate the same name is 0.07 to 0.18 (Furnas)
  - Delete, remove, expunge, wipe out, take away, ...
- A possible solution: rich *aliases* in command names
- Key concept: Design of a *congruent set* of command names
  - Up and down, right and left, add and subtract, ...
- Use of mnemonics (abbreviations)
  - Truncation, vowel deletion, etc.
  - Start with full name before introducing abbreviations
- Spelling a problem
  - But spelling checkers and correctors feasible

User must remember operators and arguments
- The issues of *syntax* (fixed order for operands or free form, operator before arguments or vice versa)
- Operator after arguments means that command termination is implicit even with a variable number of arguments
- Screen prompts can help

Example: UNIX

10.6 The role of command language syntax

Applies to non-verbal as well as verbal dialogues

Light buttons: {command argument}*
e.g.,
CIRCLE <pos1>
SQUARE <pos2>
TRIANGLE <pos3>

Paint buckets: {set_mode {arguments}*}*
e.g.,
CIRCLE <pos1> <pos2> <pos3>
SQUARE <pos4> <pos5> <pos6>

VIDEO — U of T SELECTION/POSITIONING (1981)

10.7 Query and conversational programming languages

Query language: Special-purpose language for constructing queries to retrieve information from a computerized database
Example query in several query languages (Fig. 10.1)
Query by example
Procedural vs. *non-procedural* language
Data models (hierarchical, *network*, *relational)*
Tasks (Fig. 10.2) and measures (Fig. 10.3) in evaluation

Conversational programming languages
Task language *extensible* and fully programmable
Lotus macros and the Lotus phenomenon
LOGO, APL, 4th generation languages and environments
Syntax-directed editor
Avoiding, detecting, correcting errors
Programming environments

Fig. 10.1 Queries in several query languages (Riesner, in Handbook of Human-Computer Interaction, 1988, p. 259)

Table 1: Queries in Several Query Languages

Query Languages	Example Query for "Find the names of employees in department 50"
SQL	SELECT NAME FROM EMP WHERE DEPTNO = 50
QBE	EMP | NAME: p. Brown | DEPTNO: 50 | SAL
SQUARE	EMP ('50') NAME DEPTNO
TABLET	FORM DEPTFIFTY FROM NAME. DEPTNO OF EMP KEEP ROWS WHERE DEPTNO = 50 PRINT NAME

Fig. 10.2 Some tasks used to measure ease-of-use of query languages (Riesner, in Handbook of Human-Computer Interaction, 1988, p. 261)

Table 2: Some Tasks Used to Measure Ease-of-Use of Query Languages

Task	Description
Query writing	Users are given a question stated in English and required to write a query in the given query language.
Query reading	Users are given a query written in the query language and asked to write a translation into English.
Query interpretation	Users are given a query in the query language and a printed database with data filled in. They are asked to find the data asked for by the query.
Question comprehension	Users are given an English question and a printed database and are asked to find the data asked for.
Memorization	Users are asked to memorize and reproduce a database.
Problem solving	Users are given a problem and a database and are asked to generate questions in English that would solve the problem. The questions should be answerable from the database.

Fig. 10.3 Some kinds of tests used to measure ease-of-use of query languages (Riesner, in Handbook of Human-Computer Interaction, 1988, p. 262)

Table 3: Some Kinds of Tests Used to Measure Ease-of-Use

Task	Description
Final exams of learning	These test how easy a query language is to learn; they are given at the end of teaching.
Immediate comprehension	These help identify why particular learning problems occur. They are given during teaching, immediately after some function has been taught, to determine whether subjects can use the function, given that they know it is the one to use.
Reviews	These help identify why particular learning problems occur. They are given during teaching and cover functions taught up to that time. They require that subjects know which function to use.
Productivity	These are tests of query language used by "skilled" users. They test how well the language can be used after some predetermined level of learning has been attained.
Retention	These test how easy a query language is to remember: how well it can be used by people who have been away from it for a period of time.
Relearning	These test how easy a query language is to relearn by users who have been away from it for a period of time and have forgotten some of it.

10.8 Natural language input

Some DBMS query languages are “English like” languages
    Work for limited range of discourse, subset of English

What about full natural language?
    Unlikely in foreseeable future

Habitability in restricted natural language
    “The ability of users to stay within the limits of a computer language while expressing themselves productively”
    4 domains of habitability:
        Conceptual, functional, syntactical, lexical
        Example: What is the salary of John Smith's manager?

(**Conceptual**) Not understood if no information about managers	
(**Functional**) Not understood if unable to handle that type of query	**Rephrase as:** Who is the manager of John Smith? **System:** Mary Jones What is the salary of Mary Jones?
(**Syntactical**) Not understood if can't handle possessives	**Rephrase as:** What is the salary of the manager of John Smith?
(**Lexical**) Not understood if don't know the word ”salary”	**Rephrase as:** What are the earnings of the manager of John Smith?

Problems:
    Tends to become rather verbose: many keystrokes, particularly hard on poor typists
    Problems of ambiguity, anaphora, ellipsis, etc.

Could employ voice input as well, but not necessarily

More on this topic later

10.9 Menu dialogues

Computer-initiated *display of alternatives*

Response variables
- Typing number or keyword, or hitting function key?
- Single keystroke, or ENTER required?
- Single token responses only, or arguments too?

Menu display and organization
- Menu items displayed as words or pictographs (icons)?
- Menu pages simple, pull-down, pop-up, scrolled, paged, tree structured, adaptive?
- Depth (d) versus breadth (b) tradeoff: $n = b^d$

Very deep:	b=2	d=6
Intermediate:	b=4	d=3
Shallower:	b=8	d=2
One-level:	b=64	d=1

- Generally, breadth better than depth
- Importance of menu organization:
  - Logical, alphabetic, frequency of use

Navigational aids? For example, in Lotus 1-2-3
- Hierarchical menus
- integrated help
- Command language bypass
- Extensibility

Menus can be voice menus, e.g., “Would you like to speak to... 1. Linda... 2. Susie... 3. Pierre... or 4. The operator”

VIDEO — OLYMPIC MESSAGING SYS. (IBM, 1985, SGVR 19)

10.10 Form filling dialogues

Computer-initiated *display of requirements*

Design variations
- How is cursor positioned? (Automatically, or by user?)
- How are different forms called up?
- How is help provided without obliterating form?
- One form at a time, or multiple forms in parallel?

Navigation through forms

Forms can be voice forms, as in Olympic Message System,
- e.g., “Please provide your name..... now your ID#.....”

Example of menus + forms:
- Property, or style, sheets in Xerox Star